

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Seetharaman et al.	§	
	§	Group Art Unit: 3629
Serial No. 09/966,200	§	
	§	Examiner: Araque Jr., Gerardo
Filed: September 27, 2001	§	
	§	
For: Isolating User Interface Design	§	
from Business Object Design Using	§	
Java Interface Concepts		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

37945
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on August 2, 2007.

A fee of \$510.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0457. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0457. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0457.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-18

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: None

Claims withdrawn from consideration but not canceled: None

Claims pending: 1-18

Claims allowed: None

Claims rejected: 1-18

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1-18

STATUS OF AMENDMENTS

A response to the final Office action of May 3, 2007 was filed on June 26, 2007. In the advisory action of July 24, 2007, the Examiner stated that the response was not entered.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method of developing a computer software system (specification page 3, lines 11-15). The method includes:

defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400);

defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface (specification page 3, lines 11-15; page 6, line 26 – page 7, line 5; Figure 4, reference numeral 420);

wherein the proposed view sub-system, the proposed business logic sub-system, and the proposed handler sub-system are all isolated from each other (specification page 7, lines 7-17; Figure 5);

creating the proposed view sub-system in accord with the first interface (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400); and

creating the proposed handler sub-system in accord with the second interface (specification page 3, lines 11-15; page 6, line 26 – page 7, line 5; Figure 4, reference numeral 420).

B. CLAIM 11 - INDEPENDENT

The subject matter of claim 11 is directed to a computer software system in a computer readable medium (specification page 3, lines 11-15). The computer software system includes:

first instructions defining a view sub-system including presentation objects which provide a user interface (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400);

second instructions defining a business logic sub-system including use case objects which hold business data and implement business functions (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400);

third instruction defining a handler sub-system including controller objects which control

actions of the view sub-system and actions of the business logic sub-system (specification page 3, lines 11-15; page 6, line 26 – page 7, line 5; Figure 4, reference numeral 420);

fourth instructions defining a data interface only through which the view sub-system obtains business data for the presentation objects (specification page 3, lines 17-23; page 6 lines 6-8); and

fifth instructions defining a business interface only through which the handler sub-system invokes business functions (specification page 3, lines 25-31; page 6, lines 11-15).

C. CLAIM 14 - INDEPENDENT

The subject matter of claim 14 is directed to a computer program in a computer readable medium (specification page 3, lines 11-15). The computer program product includes:

first instructions defining at least one view object including presentation objects which provide a user interface (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400);

second instructions defining at least one business logic object holding business data and implementing business functions (specification page 3, lines 11-15; page 6, lines 26-31; Figure 4, reference numeral 400);

third instructions defining at least one handler object which controls actions of at least one of the view objects and actions of at least one of the business logic objects (specification page 3, lines 11-15; page 6, line 26 – page 7, line 5; Figure 4, reference numeral 420);

fourth instructions defining a data interface only through which the at least one view object obtains business data for the presentation objects (specification page 3, lines 17-23; page 6 lines 6-8); and

fifth instructions defining a business interface only through which the at least one handler object invokes business functions (specification page 3, lines 25-31; page 6, lines 11-15).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUND OF REJECTION 1

The sole ground of rejection is whether the Examiner failed to state a prima facie obviousness rejection against claims 1-18 under 35 U.S.C. § 103 in view of *Beckett et al.*, System and Method For Visual Application Development Without Programming, U.S. Patent 6,564,368, May 13, 2003, (hereinafter “*Beckett*”).

ARGUMENT

I. Rejection and Legal Standards

The sole ground of rejection is whether the Examiner failed to state a prima facie obviousness rejection against claims 1-18 under 35 U.S.C. § 103 in view of *Beckett et al.*, System and Method For Visual Application Development Without Programming, U.S. Patent 6,564,368, May 13, 2003, (hereinafter “*Beckett*”). As proved below, this rejection is clearly in error. Therefore, Applicants request that the Board of Patent Appeals and Interferences reverse the rejection and direct the Examiner to allow the claims.

Claim 1 is a representative claim of this grouping of claims. Applicants also argue claim 2, below. In regards to claim 1, the Examiner asserts the following:

In regards to **claim 1**, *Beckett* discloses a method of developing a computer system, comprising the computer-implemented steps of:

defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface (**Column 1 Lines 24-30, 44-47; Column 3 Lines 1-12, 44-47**);

defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler subsystem and the proposed business logic sub-system interact only via the second interface (**Column 1 Lines 24-30, 44-47; Column 3 Lines 1-12, 44-47**);

wherein the proposed view sub-system, the proposed business logic sub-system, and the proposed handler sub-system are all isolated from each other (**Column 1 Lines 44-47; Column 3 Lines 1-12, 44-47**);

creating the proposed view sub-system in accord with the first interface (**Column 6 Lines 20-27**); and

creating the proposed handler sub-system in accord with the second interface (**Column 6 Lines 20-27**).

Beckett, however, fails to explicitly state the exact arrangement of 3 sub-systems with interfaces between each sub-system.

However, *Beckett* does disclose that multiple interfaces can be used to connect multiple objects and that one of ordinary skill in the art would know

that there are numerous ways of connecting the sub-systems (**Column 6 Lines 20-27; Column 8 Lines 23-27**). Moreover, it would have been obvious that the sub-systems would be isolated from each other when an interface is placed between them. Further still, it would be obvious that the sub-systems would be in accord with their respected interfaces in order to avoid compatibility issues.

Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention that using the teachings of *Beckett* (**specifically Col. 6 L. 20-27 & Col. 8 L. 23-27**) the industry is assured the rapid, high-quality construction of products.

Final office action of May 3, 2007 pp. 2-3 (emphasis in original).

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

II. The Examiner Failed to State a *Prima Facie* Obviousness Rejection Because *Beckett* Does Not Teach or Suggest the Features of Claim 1 in the Manner Asserted by the Examiner

II.A. Claim 1

Regarding claim 1, the Examiner has failed to state a *prima facie* obviousness rejection because the proposed modification of *Beckett* does not teach or suggest all of the features of claim 1 in the manner asserted by the examiner. Claim 1 is as follows:

1. A method of developing a computer software system, comprising the computer-implemented steps of:
 - defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface;
 - defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface;
 - wherein the proposed view sub-system, the proposed business logic sub-system, and the proposed handler sub-system are all isolated from each other;
 - creating the proposed view sub-system in accord with the first interface; and
 - creating the proposed handler sub-system in accord with the second interface.

Beckett does not teach or suggest a number of features of this claim. For example, *Beckett* does not teach the features of, “defining a *first* interface associated with a proposed view sub-system and with a proposed business logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface,” and *Beckett* does not teach the features of, “defining a *second* interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface,” as recited in claim 1. The Examiner erroneously asserts otherwise, citing to the following portions of *Beckett*:

Programs are created from instructions of a programming language that are accumulated into the program's source code. The source code controls presentation, interfaces, and logic. A programmer authors source-code and compiles it into processor machine code utilizing a compiler compatible with both the source code's language and target processor.

Beckett, col. 1, lines 24-30.

Applications are constructed from one or more programs. Programmers write source code leveraging interfaces that enable disparate programs to

interact with each other and provide greater utility.

Beckett, col. 1, lines 44-47.

These connections between classes are defined within a visual environment. The relationships can be programmatically attached by name to object instances during program execution. Because these relationships are stored in a resource and are dynamically bound by name to the objects, they can be created and modified without requiring the source code of the objects being associated to be changed. This eliminates hard coded dependencies between objects that impede reuse of the objects in other contexts. This type of program requires meta-data, full dynamic binding and probing support in the objects being connected with the invention.

Beckett, col. 3, lines 1-12.

Therefore, the present invention permits business logic, data translations, expressions, and other algorithms to be visually modeled using the interface manager and its dynamic properties as well as the Connection Editor.

Beckett, col. 3, lines 44-47.

Beckett teaches a method for visual application development without programming. The above cited portions state that programs are created from instructions of a programming language that are accumulated into the program's source code. The source code determines the presentation, interfaces, and logic. The source code is then compiled into machine code. Disparate programs then interact with each other through interfaces to form an application. These connections between classes are defined within a visual environment without requiring the source code of the objects being associated to be changed.

Beckett's method for visual application development without programming does not teach or suggest the features of claim 1. *Beckett* broadly states that business logic, data translations, expressions, and other algorithms can be visually modeled using the interface manager. Applications are constructed by connecting the properties of desired programs using the Connection Editor graphically without any source code programming. *Beckett*, col. 3, ll. 25-50. However, *Beckett* does not teach the claimed feature of, "defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system" as recited in claim 1. *Beckett's* disclosure makes no reference to defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system. Thus, *Beckett* also does not suggest defining a first interface associated with a proposed view sub-system and with a proposed business

logic sub-system as recited in claim 1.

Additionally, *Beckett* does not teach the second step of, “defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface” as recited in claim 1. Again, *Beckett* broadly states that business logic, data translations, expressions, and other algorithms can be visually modeled using the interface manager. However, *Beckett* makes no reference to defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system. Furthermore, *Beckett* makes no mention of a handler subsystem anywhere in *Beckett’s* disclosure. Therefore, *Beckett* also does not teach or suggest this feature as recited in claim 1.

In addition *Beckett* does not teach the claimed feature, “wherein the proposed view sub-system, the proposed business logic sub-system, and the proposed handler sub-system are all isolated from each other.” The Examiner mistakenly cites the above quoted passages of *Beckett* as teaching this feature. However, neither the above portion nor any other portion of *Beckett* teaches a proposed handler sub-system. Consequently, *Beckett* also does not teach or suggest this feature as recited in claim 1.

Moreover, because *Beckett* does not teach or suggest the first three features of claim 1, *Beckett* inherently also does not teach or suggest the remaining features of, “creating the proposed view sub-system in accordance with the first interface and creating the proposed handler sub-system in accordance with the second interface.” Accordingly, under the standards of *In re Royka*, the Examiner has failed to state a prima facie obviousness rejection against claim 1 or any other claim in this grouping of claims.

II.B. Claim 2

Furthermore, claims 2-10, 12, 13, and 15-18 recite features not taught or suggested by *Beckett*. For example, claim 2 states defining a third interface associated with the proposed view sub-system and with the proposed handler sub-system and creating the proposed view sub-system in accord with both the first and third interfaces. As shown in the above-quoted portions of *Beckett*, *Beckett* is devoid of disclosure regarding defining a third interface. Even if *Beckett* showed defining a third interface, *Beckett* does not teach creating the proposed view sub-system *in accord with both the first and third interfaces*, as required by claim 2. Therefore, the proposed

modification of *Beckett* fails to teach or suggest all of the features of claim 2. Accordingly, under the standards of *In re Lowry*, the Examiner has failed to state a prima facie obviousness rejection against claim 2.

III. The Examiner Failed To State a Proper Reason To Modify *Beckett* under the Standards of *KSR Int'l.*

Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

In regards to a reason to modify *Beckett*, the Examiner states that:

Beckett, however, fails to explicitly state the exact arrangement of 3 sub-systems with interfaces between each sub-system. However, *Beckett* does disclose that multiple interfaces can be used to connect multiple objects and that one of ordinary skill in the art would know that there are numerous ways of connecting the sub-systems (Column 6 Lines 20-27; Column 8 Lines 23-27). Moreover, it would have been obvious that the sub-systems would be isolated from each other when an interface is placed between them. Further still, it would be obvious that the sub-systems would be in accord with their respected interfaces in order to avoid compatibility issues.

Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention that using the teachings of *Beckett* (specifically Col. 6 L. 20-27 & Col. 8 L. 23-27) the industry is assured the rapid, high-quality construction of products.

Final office action of May 3, 2007, p. 3.

The Examiner has not stated an *articulated reasoning with some rational underpinning* to support the legal conclusion of obviousness. Instead, the examiner has only asserted a purported advantage to modifying *Beckett*, *without any reasoning whatsoever tying the purported advantage to the legal conclusion of obviousness*. Instead, the examiner relies on a conclusory, unsupported statement that “the industry is assured the rapid, high-quality construction of products” and then

assumes that somehow this nebulous statement somehow results in the *legal conclusion* of obviousness. The Court in *KSR Int'l.* specifically forbids this kind of unsupported, conclusory reasoning. Therefore, the examiner failed to state a proper reason to combine the references under the standards of *KSR Int'l.* Accordingly, the examiner failed to state a prima facie obviousness rejection against claim 1 or any other claim in this grouping of claims.

Additionally, despite the Examiner's attempt to state the contrary, the Examiner has implicitly admitted that no pre-existing reason exists to modify *Beckett* to achieve the invention of claim 1. The Examiner states that, "*Beckett*, however, fails to explicitly state the exact arrangement of 3 sub-systems with interfaces between each sub-system." The Examiner then states, "that one of ordinary skill in the art would know that there are *numerous ways* of connecting the sub-systems" (emphasis supplied).

Assuming, *arguendo*, that these and other statements made by the Examiner are accurate – which they are not – the Examiner's statements on their face prove that no reason exists to modify *Beckett* to achieve the invention of claim 1. Because one of ordinary skill knows that *numerous ways* exist to connect subsystems, one of ordinary skill also knows that ***no reason exists to assume that any one particular combination of connections should be adopted over another combination.*** For this reason, no basis exists to assume that the particular arrangement of the three sub-systems in claim 1 should be adopted solely in view of *Beckett's* disclosure. Therefore, the Examiner has implicitly admitted that no reason exists to modify *Beckett* to achieve the invention of claim 1 under the standards of *KSR Int'l.*

Additionally, the Examiner offers no reason why one of ordinary skill would find the *particular claimed combination* obvious in view of *Beckett*. Instead, the Examiner only offers a nebulous statement regarding the "obvious" presence or arrangement of the claimed features in view of *Beckett*. In view of the fact that the Examiner provides no basis whatsoever that the particular claimed arrangement would be obvious, the Examiner has only made a conclusory statement regarding the obviousness of claim 1 in view of the cited reference. The Court in *KSR Int'l.* explicitly states that conclusory statements are insufficient to establish the *legal conclusion* of obviousness.

Furthermore, the Examiner has asserted an overly-broad reason as a motivation to modify *Beckett*. The Examiner states that modifying *Beckett* to achieve the invention of claim 1 would be

obvious because, “the industry is assured the rapid, high-quality construction of products.” However, this statement is overly-broad to serve as a proper teaching, suggestion, or motivation to modify *Beckett* to achieve *the specific invention of claim 1*. For example, the Examiner’s proposed reason to modify *Beckett* is already achieved by *Beckett*. *Beckett* already provides a method for “assuring” rapid, high-quality construction of products. Thus, under the Examiner’s reasoning, *Beckett* is already complete and one of ordinary skill would have no reason to modify *Beckett* to achieve *the specific invention of claim 1*.

The Examiner has offered no *specific* reason to modify *Beckett* to achieve the *specific* invention of claim 1. Instead, the Examiner has made generalizations that are too broad to provide a reason why one of ordinary skill would find claim 1 obvious in view of *Beckett*. Thus, the Examiner has failed to state a proper reason to modify *Beckett* to achieve the invention of claim 1 under the standards of *KSR Int’l*. Accordingly, the Examiner has failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

IV. *Beckett* Teaches Away from the Invention of Claim 1

A reference may be said to “teach away” from the claimed invention when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant. *In re Gurley*, 27 F.3d 551, 553, 31 U.S.P.Q.2D 1130, 1131 (Fed. Cir. 1995).

Under this standard, *Beckett* teaches away from the invention of claim 1. *Beckett* teaches at col. 3 lines 30-43 that the “...Connection Editor interacts with the interface manager of all programs in the system to render that real time status of connections between disparate program interfaces Changes in any interface property during run time operation are propagated to all other connected interface properties.”

Further at col. 6 lines 22-49 *Beckett* teaches that, “...furthermore, each component must be able to interact with any other component in order to carry out connections ... this is accomplished by requiring that each component implement a standard interface mechanism ... the Connection Editor 203 has a common mechanism to interact with all programs ... an Interface Manager 410. At col. 8 lines 17-20 *Beckett* teaches “... Interface Manager 410 only requires a reference to another components’ interface manager ... to establish connections.”

Beckett teaches the use of a “Connection Editor interacts with the interface manager of all programs ...” in a standardized manner to achieve the results of *Beckett*. Further, *Beckett* teaches connections be used in a standard manner to allow propagation of information as noted previously at col. 3 lines 30-43, “...Connection Editor interacts with the interface manager of all programs in the system to render that real time status of connections between disparate program interfaces Changes in any interface property during run time operation are propagated to all other connected interface properties.”

In contrast, the claimed invention isolates sub-systems through the use of the claimed interfaces. Further, as claimed, creation of the sub-systems is in accordance with a respective interface and not a standard interface technique as taught by *Beckett*. Therefore, *Beckett* teaches the use of standard interface techniques.

Further, *Beckett* defines the interfaces to the programs at col. 6 lines 22-49, which provides, “... create the proposed view sub-system in accordance with the first interface ...”, in which the interface definitions determine the sub-system creation. Thus, the teachings of *Beckett* are backwards to the explicitly recited features of claim 1.

Because *Beckett* teaches a technique that is backwards to the invention of claim 1, a person of ordinary skill would be led in a direction divergent from the path taken by claim 1. Similarly, because *Beckett* teaches the use of standard interface techniques, as opposed to the different interface techniques explicitly recited in claim 1, one of ordinary skill would be discouraged from following the path set out in *Beckett*. Therefore, under the standards of *In re Gurley*, *Beckett* teaches away from the invention of claim 1. Accordingly, the examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

V. No Reason Exists to Further Modify *Beckett* to Achieve the Invention of Claim 1

A *prima facie* case of obviousness cannot be properly based upon a prior art reference if the prior art reference requires a modification that destroys the intended purpose or function of the claimed invention. In the case at hand, one skilled in the art could not modify the operation of *Beckett*, defining interfaces to programs, to become the creation of sub-systems after interfaces have been defined as claimed, without vitiating the purpose and teachings of *Beckett*. Accordingly, no reason exists under the standards of *KSR Int'l.* to modify *Beckett* to achieve the legal conclusion that claim 1 is obvious. Hence, the examiner failed to state a *prima facie* obviousness rejection

against claim 1 or any other claim in this grouping of claims.

VI. Rebuttal to the Examiner's Response

In response to the above facts, the examiner makes several incorrect assertions. Applicants address each in turn.

The examiner first states the following:

22. Applicants argue that *Beckett* does not teach, "defining a first interface associated with a proposed view sub-system and with a proposed logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface and defining a second interface associated with a proposed handler sub-system and with the proposed logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface." As a further note, the Examiner understands the applicant's invention to only be various interfaces with various sub-systems, or objects, wherein each interface contains a type of sub-system and that each sub-system is isolated from one another. The sub-systems use their respective interface to communicate to one another and to other parts of the system. The Examiner asserts that although *Beckett* does not disclose the exact arrangement or the same title of each of the sub-systems, as disclosed by the applicant, *Beckett* does disclose that multiple interfaces can be used with their respective objects (**Column 6 Lines 20 - 27**). Each object communicates to one another through some interface.

Final office action of May 3, 2007, p. 9 (emphasis in original).

The examiner incorrectly believes that Applicant's invention to "only be" various interfaces with various sub-systems or objects, wherein the sub-systems are isolated from one another. This opinion of the examiner is fundamentally incorrect. Claim 1 requires not some nebulous set of interfaces that are isolated from each other, but rather a *particular arrangement of interfaces*. For example, claim 1 requires, "defining a first interface associated with a *view sub-system* and with a *proposed business logic sub-system*." In the next clause claim 1 requires, "defining a second interface associated with a *proposed handler sub-system* and with the *proposed business logic sub-system*." Thus, the first interface and the second interface define a *specific relationship* among the first and second interfaces and the three sub-systems, wherein both interfaces interact with one of the sub-systems.

Thus, the examiner's statement that Applicant's invention to "only be" various interfaces in the manner asserted by the examiner *ignores the features of the claims* and also belies the examiner's complete misunderstanding of claim 1. For this reason, the examiner's statements not only fail to bolster the examiner's argument, but rather actually show that the basic rejection is fundamentally flawed.

The examiner also states that, "The Examiner asserts that although *Beckett* does not disclose the exact arrangement or the same title of each of the sub-systems, as disclosed by the applicant, *Beckett* does disclose that multiple interfaces can be used with their respective objects." However, the examiner has no basis to assume that any particular arrangement of interfaces in *Beckett* would be obvious to one of ordinary skill. Certainly, the examiner has provided no technical reason why the particular arrangement in claim 1 would be obvious in view *Beckett*. In fact, under the examiner's reasoning, *absolutely any* arrangement of interfaces would be obvious in view of *Beckett*.

However, this reasoning fails to comport with the standards of *KSR Int'l.*, which requires that the examiner engage in a reasoned analysis of how claim 1 is obvious in view of the cited reference. The examiner has provided no such reasoning, but rather has only cited a purported, overly-broad advantage and then assumed the conclusion of obviousness. Again, this reasoning fails to comport with the standards of *KSR Int'l.* Therefore, the examiner failed to state a prima facie obviousness rejection against claim 1 or any other claim in this grouping of claims.

Nevertheless, the examiner also states that:

Moreover, *Beckett* also discloses that,

"The Connection Editor 203 shows the status of connections between programs and allows end-users to create connections between programs
(Column 5 Lines 23 - 25).

Thus, Interface Manager 410 only requires a reference to another components interface manager and the name of the connected interface property as the minimum information to establish a connection between interface properties. With this information, the information managers of each component can automate data flow between the components without programming. One ordinarily skilled in the art would know that this is just one of numerous ways that a connection editor-or any program capable of querying data from class metadata, internal-storage, or external storage-could query available connection points from a program (Column 8 Lines 17

Final office action of May 3, 2007, pp. 9-10.

The Examiner states that, "one of ordinary skill in the art would know that this [flow of components] is just one of numerous ways that a connection editor...could query available connection points from a program." However, again, the examiner proceeds from the false assumption that any particular combination would be obvious in view of *Beckett*. However, the *legal conclusion* of obviousness requires that the examiner engage in the analysis set forth in *KSR Int'l*. As shown above, the examiner has not engaged in this analysis to show *why the particular implementation in claim 1* would be obvious in view of the cited references.

In an analogy, the examiner is effectively stating that a particular house constructed with a hammer is obvious because hammers are known in the art. Under the examiner's logic, one of ordinary skill would conclude that *any particular structure of the house* would be "obvious," no matter how ingenious the structure, simply because a patent directed to a hammer states that hammers can be used in the construction of houses.

When placed in these terms, the flawed nature of the examiner's reasoning becomes apparent. The analogy is also fully apt, because the "hammer" or "tool" in *Beckett* is the connection editor cited by the examiner. The *particular structure of the house* corresponds to the *detailed and explicit interrelationship of the features of claim 1*.

Thus, the examiner's reasoning attempts to establish obviousness by simply stating that the *exact features in claim 1 are obvious* because a purported tool for building connections exists and those of ordinary skill purportedly know how to use the tool. This assertion is wrong. Thus, the examiner's observations and reasons underpinning the rejection are wrong. Hence, under the standards of *KSR Int'l*, the rejection is wrong. Accordingly, the rejection should be overturned.

Nevertheless, the examiner also states that:

Applicant's argument that *Beckett* does not teach defining interfaces and subsystem is incorrect. The step of defining each of these components has obviously already been done in order for *Beckett* to carry out the invention, i.e. the programming of these components has already been done. *Beckett* teaches communication of these components with one another and that one skilled in the art would know that there are numerous methods of associating each of these devices depending on the client's needs as well as the amount of resources available.

Final office action of May 3, 2007, p. 10.

Again, the examiner relies on the fundamentally incorrect premise that one of ordinary skill would find a particular implementation obvious over the *millions* of possible combinations available through the use of the tool described in *Beckett*. Thus, for the reasons given above, the examiner continues to fail to state a prima facie obviousness rejection in view of *Beckett*.

Additionally, the examiner misunderstands Applicant's arguments. Instead of responding to the fact that *Beckett* does not teach the interrelation of the claim features, as shown above, the examiner simply states that, "Applicant's argument that *Beckett* does not teach defining interfaces and subsystem is incorrect." However, in making this simplification the examiner has oversimplified Applicant's statements and then responded to a fundamentally incorrect premise. Applicants do not assert that *Beckett* does not teach defining interfaces and subsystems, as opined by the examiner. Applicants have *proved* that *Beckett* does not teach or suggest the explicit language required in claim 1, particularly with respect to the interrelation of the claimed features. Therefore, under the standards of *In re Royka*, the examiner failed to state a prima facie obviousness rejection against claim 1 or any other claim in this grouping of claims.

VII. CONCLUSION

As shown above, the Examiner has failed to state valid rejections against any of the claims. Therefore, Applicants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Applicants request that the Board direct the Examiner to allow the claims.

/Theodore D. Fay III/
Theodore D. Fay III
Reg. No. 48,504
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method of developing a computer software system, comprising the computer-implemented steps of:

defining a first interface associated with a proposed view sub-system and with a proposed business logic sub-system, wherein the proposed view sub-system and the proposed business logic sub-system interact only via the first interface;

defining a second interface associated with a proposed handler sub-system and with the proposed business logic sub-system, wherein the proposed handler sub-system and the proposed business logic sub-system interact only via the second interface;

wherein the proposed view sub-system, the proposed business logic sub-system, and the proposed handler sub-system are all isolated from each other;

creating the proposed view sub-system in accord with the first interface; and

creating the proposed handler sub-system in accord with the second interface.

2. The method according to claim 1, further comprising the steps of:

defining a third interface associated with the proposed view sub-system and with the proposed handler sub-system; and

creating the proposed view sub-system in accord with both the first and third interfaces.

3. The method according to claim 1, further comprising the steps of:

defining a fourth interface associated with the proposed view sub-system and with the

proposed handler sub-system; and

creating the proposed handler sub-system in accord with both the second and the fourth interfaces.

4. The method according to claim 1, further comprising the steps of:

defining a third interface associated with the proposed view sub-system and the proposed handler sub-system;

defining a fourth interface associated with the proposed view sub-system and with the proposed handler sub-system;

creating the proposed view sub-system in accord with both the first and third interfaces;
and

creating the handler sub-system in accord with both the second and the fourth interfaces.

5. The method according to claim 1, wherein:

the first interface defines a plurality of methods for data storage and retrieval that are implemented in the business logic sub-system.

6. The method according to claim 1, wherein:

the second interface defines a plurality of methods of business logic that are implemented in the business logic sub-system.

7. The method according to claim 2, wherein:

the third interface is a listener interface that defines a plurality of methods in the handler sub-system, and wherein the listener interface responds to actions in the view sub-system.

8. The method according to claim 3, wherein:

the fourth interface defines a plurality of methods which are implemented in the view sub-system for use by the handler sub-system.

9. The method according to claim 1, wherein:

the view sub-system includes a plurality of user interface objects;

the handler sub-system includes a plurality of use case control objects; and

the business logic sub-system includes a plurality of business logic objects.

10. The method according to claim 1, wherein:

the sub-systems are created independently of each other once the interfaces have been defined.

11. A computer software system in a computer readable medium, said system comprising:

first instructions defining a view sub-system including presentation objects which provide a user interface;

second instructions defining a business logic sub-system including use case objects which hold business data and implement business functions;

third instruction defining a handler sub-system including controller objects which control

actions of the view sub-system and actions of the business logic sub-system;

fourth instructions defining a data interface only through which the view sub-system obtains business data for the presentation objects; and

fifth instructions defining a business interface only through which the handler sub-system invokes business functions.

12. The system according to claim 11, further comprising:

sixth instructions defining a listener interface through which the handler sub-system responds to events in the user interface.

13. The system according to claim 11, further comprising:

sixth instructions defining a view action interface through which the handler sub-system invokes actions in the user interface.

14. A computer program in a computer readable medium, said program comprising:

first instructions defining at least one view object including presentation objects which provide a user interface;

second instructions defining at least one business logic object holding business data and implementing business functions;

third instructions defining at least one handler object which controls actions of at least one of the view objects and actions of at least one of the business logic objects;

fourth instructions defining a data interface only through which the at least one view object obtains business data for the presentation objects; and

fifth instructions defining a business interface only through which the at least one handler object invokes business functions.

15. The computer program according to claim 14, further comprising:

sixth instructions defining a listener interface through which the handler object responds to events in the user interface.

16. The computer program according to claim 14, further comprising:

sixth instructions defining a view action interface through which the handler object invokes actions in the user interface.

17. The computer program system of claim 11 further comprising:

sixth instructions for further defining the view sub-system, the business logic sub-system, and the handler sub-system such that each sub-system is isolated from another sub-system.

18. The computer program of claim 14 further comprising:

sixth instructions for further defining the view sub-system, the business logic sub-system, and the handler sub-system such that each sub-system is isolated from another sub-system.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.